ULTIMATE SQL + ANALYTICS INTERVIEW REVISION (WORKSTUDENT DATA ANALYST)

This document is designed to help you confidently pass a Data Analyst / Werkstudent interview in Germany.

Covers: SQL, Joins, Aggregations, Subqueries, Case, Business Logic, Interview Patterns, Analyst Mindset, and Key Concepts.

SECTION 1 — SQL FOUNDATIONS

----

1. SELECT BASICS

- SELECT column1, column2 FROM table;

- SELECT * FROM table;

- DISTINCT keyword removes duplicates.

2. FILTERING (WHERE)

- WHERE column = value

- WHERE column > value

- WHERE column IN (...)

- WHERE column BETWEEN x AND y

- WHERE column LIKE '%abc%'

- WHERE column IS NULL / IS NOT NULL

3. ORDER & LIMIT

- ORDER BY column ASC

- ORDER BY column DESC

- LIMIT n

SECTION 2 — AGGREGATIONS (CORE INTERVIEW PART)

----

4. Common Aggregate Functions:

- SUM(column)
- COUNT(*)
- AVG(column)
- MIN(column)
- MAX(column)

5. GROUP BY (IMPORTANT)

- All non-aggregated columns must be in GROUP BY.

Example:

```
SELECT customer_id, SUM(amount)
FROM orders
GROUP BY customer_id;
```

6. HAVING (AFTER GROUPING)

- HAVING SUM(amount) > 500
- WHERE cannot use aggregates, HAVING can.

SECTION 3 — JOINS (THE INTERVIEW BACKBONE)

----

7. INNER JOIN

Returns matching rows.

```
SELECT *
FROM orders o
JOIN customers c ON o.customer_id = c.customer_id;
```

8. LEFT JOIN

Include all from left table, even if no match.

9. RIGHT JOIN

Include all from right table.

10. FULL OUTER JOIN

If supported.

11. JOIN Best Practices

- Always join on correct key.

- Use table aliases (c, o, p).

SECTION 4 — STRING FUNCTIONS

----

12. CONCATENATION

CONCAT(first_name, ' ', last_name) AS full_name

13. UPPER(), LOWER(), LENGTH()

SECTION 5 — CASE STATEMENTS (BUSINESS LOGIC)

----

14. CASE WHEN

Create categories/labels:

```
CASE
WHEN SUM(amount) > 1000 THEN 'High'
WHEN SUM(amount) BETWEEN 500 AND 1000 THEN 'Medium'
ELSE 'Low'
END AS spending_level
```

SECTION 6 — SUBQUERIES (REAL INTERVIEW LEVEL)

----

15. Subquery in WHERE

Filter based on aggregated value:

```
SELECT *
FROM orders
WHERE amount > (SELECT AVG(amount) FROM orders);
```

16. Subquery in SELECT

Show global statistic with every row:

```
SELECT amount,
(SELECT AVG(amount) FROM orders) AS avg_amount
FROM orders;
```

17. Subquery in FROM (Derived Table)

MOST IMPORTANT: use when you need to aggregate and then filter.

```
SELECT t.customer_id, t.total_spending
FROM (
SELECT customer_id, SUM(amount) AS total_spending
FROM orders
GROUP BY customer_id
) AS t
WHERE t.total_spending > 500;
```

SECTION 7 — BUSINESS ANALYSIS SQL PATTERNS

----

18. Total Revenue

```
SELECT SUM(amount) AS total_revenue FROM orders;
```

19. Revenue by Customer

```
SELECT customer_id, SUM(amount)
```

```
FROM orders
GROUP BY customer_id;
```

20. Revenue by Country

```
SELECT c.country, SUM(o.amount)
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.country;
```

21. Top Spenders (Ranking Logic)

```
SELECT customer_id, SUM(amount)
FROM orders
GROUP BY customer_id
ORDER BY SUM(amount) DESC
LIMIT 5;
```

22. Monthly Trends

```
SELECT DATE_TRUNC('month', order_date) AS month,
SUM(amount)
FROM orders
GROUP BY month
ORDER BY month;
```

23. Customer Order Count

```
SELECT customer_id, COUNT(*) AS num_orders
FROM orders
GROUP BY customer_id;
```

SECTION 8 — ANALYTICS MINDSET (WHAT INTERVIEWERS CHECK)

----

24. Can you translate business questions into SQL?

Examples:

- "Who are top customers?"
- "Which category drives most revenue?"
- "Is revenue growing over time?"
- "Which market is low performing?"

25. Can you read messy data and clean it mentally?

26. Do you know why we use GROUP BY vs WHERE vs HAVING?

27. Do you understand JOIN logic strongly?

28. Can you explain queries in simple English?

(EVERY interviewer checks this.)

SECTION 9 — COMMON PITFALLS TO AVOID

----

29. Never write aggregate in WHERE.

Use HAVING instead.

30. GROUP BY all non-aggregated columns.

31. Correct JOIN key is crucial.

32. Aliases cannot be used inside HAVING (usually).

33. Use commas only in:

- SELECT columns
- GROUP BY columns
- ORDER BY columns

SECTION 10 — MINI INTERVIEW CHEATSHEET

• SELECT → Choose columns

• FROM → Source table

• JOIN → Combine tables

• WHERE → Filter before grouping

• GROUP BY → Aggregate breakout

• HAVING → Filter after grouping

• ORDER BY → Sorting

• LIMIT → Restrict results

SECTION 11 — FINAL INTERVIEW TIPS

✓ Speak your logic aloud (shows clarity)

✓ Always mention assumptions

✓ Use aliases

✓ Use CASE to show business insight

✓ Use subqueries only when needed

✓ Understand the relationship between tables

✓ Ask clarifying questions

✓ Present insights, not just numbers

SECTION 12 — YOUR CONFIDENCE MESSAGE

You now have:

- SQL fundamentals

- Aggregations
- Joins
- CASE logic
- Subqueries (Level 1–3)
- Business patterns
- Interview-ready mindset

This is enough to PASS any Data Analyst / Werkstudent interview confidently.

You are ready.

END OF DOCUMENT
